一种基于动态惯性权重的鸟群优化算法

高宏进，王　力

(贵州大学 大数据与信息工程学院，贵阳 550025)

摘　要：鸟群算法（BSA）作为一种新型的元启发式群智能算法，存在易陷入局部最优、收敛速度慢和求解精度低等问题。针对原鸟群算法在求解最优化问题中的不足，提出一种基于动态惯性权重的鸟群优化算法(DBSA)。该算法通过引入非线性动态惯性权重修正鸟群飞行间隔，平衡种群全局搜索与局部搜索能力；在模拟鸟群生产者觅食的过程中引入莱维飞行，替换原算法中生产者的觅食策略提高算法活力和有效性。实验表明改进后的鸟群算法有效提高了算法的收敛速度和寻优精度。

关键词：鸟群算法；函数优化；动态惯性权重；莱维飞行

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.11.0784

Bird swarm alogrithm based on dynamic inertia weight

Gao Hongjin, Wang Li

(*School of Big Date & Information Engineering Guizhou University*, *Guiyang* 550025, *China*)

Abstract: As a new kind of heuristic swarm intelligence algorithm, the Bird Swarm Algorithm (BSA) easily falls into problems about local optimal, slow convergence speed and low resolution accuracy. Considering the fact that the original Bird Swarm Algorithm is not sufficient to solve the issue in terms of optimization, this paper proposes an optimization algorithm, Dynamic Inertia Weight-Bird Swarm Algorithm (DBSA) . The algorithm corrects birds flying interval by introducing nonlinear dynamic inertia weight, balancing the abilities of population global search and local search; this paper introduces the parameter of levy flight in the process of simulation of the foraging birds producer, advancing algorithm's vitality and effectiveness via replacing the original algorithm producers foraging strategies. As a result, experiments show that the modified algorithm improves the convergence speed and optimization accuracy effectively.

Key words: birds swarm algorithm; Function optimization; dynamic inertia weight; Levy flight

0　引言

鸟群算法[1]（bird swarm alogrithm，BSA）是由 Meng 等人于 2015 年提出的一种基于鸟群行为的群智能优化算法，其思想源于鸟群的三个主要群体行为，即飞行行为、觅食行为和警戒行为[2]，BSA 算法依赖飞行行为跳出局部最优进行全局搜索，通过觅食行为记录个体和群体最好的解，通过觅食行为和警戒行为的随机切换搜索当前局部的最优解，而通过飞行间隔 FQ 来平衡算法全局搜索和局部搜索的能力[3]。BSA 算法不仅具有运算速度快、鲁棒性好等优点，而且在解决多邻域最优化问题时具有明显优势[4-5]。此外改进算法提高算法性能也成为一个重要的研究领域，本文针对 BSA 算法在求解复杂函数时易陷入局部最优、收敛速度慢和寻优精度低等问题，提出一种基于动态惯性权重的鸟群优化算法（bird swarm algorithm based on dynamic inertia weight ，DBSA ），通过引入非线性动态惯性权重修正鸟群飞行间隔，从而平衡算法局部搜索与全局搜索之间关系；通过引入莱维飞行提高算法活力及有效性，克服早熟。实验结果表明，DBSA 算法能够有效提高算法的收敛速度和寻优精。

1　经典鸟群算法

鸟群算法是对鸟群群体行为和群体互动的简化，它模仿的是鸟群的飞行行为、觅食行为和警戒行为，并用这种群体智慧解决最优化问题[6-7]。鸟群算法可以被如下规则简化描述[8]：

a) 鸟群中的每只鸟可以在觅食行为和警戒行为之间随机切换。

b) 觅食期间，每只鸟可以迅速记录和更新个体和群体之间最好的觅食经验，这个经验将用于觅食，群体信息将即刻共享于整个鸟群。

c）保持警戒的时候，每只鸟都试图向群体中心靠近，而具

收稿日期：2017-11-23；**修回日期：**2018-02-04

作者简介：高宏进（1994-），男，山东济宁人，硕士研究生，主要研究方向为数字图像处理（1364355636@qq.com）；王力（1971-），男，贵州安顺人，教授，主要研究方向为机器学习、神经网络.

有较高食物储量的鸟类更倾向于向群体中心靠近。

d) 鸟类可以周期性地向其他位置移动。当移动到另一个位置时，鸟类一般要在生产者和乞讨者之间作出选择，具有最高食物储量的鸟将成为生产者，而具有最低食物储量的鸟将成为乞讨者，其他鸟将随机选择成为生产者和乞讨者。

e)生产者积极寻觅食物，乞讨者随机跟随一个生产者寻找食物。

经典鸟群算法流程图如图 1 所示。

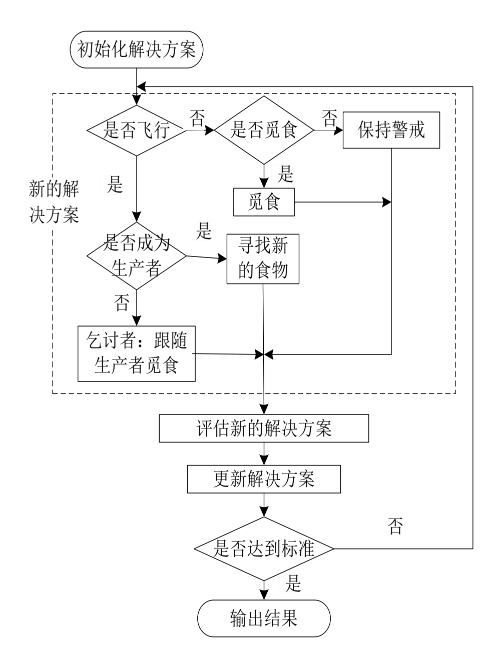


图 1　经典鸟群算法流程图

假设有 N 只鸟在 D 维空间中飞行觅食，第 i 只鸟在 t 时刻的位置描述为

$$x_i^t\left(i\in[1,2,3,\cdots,N]\right) \tag{1}$$

基于规则 a)，每只鸟的决策依赖于一个[0,1]的随机数，同时设定一个常量 P，当随机数小于 P 时，该鸟将进行觅食，否则，保持警戒[8]。

基于规则 b)，鸟群的觅食行为遵从每只鸟自身的经验以及种群经验，觅食行为的位置更新公式为

$$x_{i,j}^{t+1}=x_{i,j}^t+\left(p_{i,j}-x_{i,j}^t\right)\times C\times rand\left(0,1\right)+\left(g_i-x_{i,j}^t\right)\times S\times rand(0,1) \tag{2}$$

其中：$j\in[1,\cdots,D]$，$\mathrm{rand}(0,1)$ 表示在[0,1]均匀分布的随机数，C 和 S 是两个正数，分别称为感知系数和社会加速系数。$p_{i,j}$ 表示第 i 只鸟早先的最佳位置，g_i 表示群体共享的早先最佳位置[9]。

基于规则 c)，鸟群保持警戒时尝试向种群中心移动，并伴随着与同类的竞争，因而不能直接向种群中心移动，警戒行为的位置更新公式为

$$x_{i,j}^{t+1}=x_{i,j}^t+A_1\left(mean_j-x_{i,j}^t\right)\times r\,\mathrm{and}\left(0,1\right)+A_{21}\left(p_{i,j}^t-x_{i,j}^t\right)\times r\,\mathrm{and}(-1,1) \tag{3}$$

$$A_1=a_1\times\exp\left(-\frac{pFit_i}{sumFit+\varepsilon}\times N\right) \tag{4}$$

$$A_2=a_2\times\exp\left(-\frac{pFit_i-pFit_k}{\left|pFit_k-pFit_i\right|+\varepsilon}\times N\times\frac{pFit_k}{sumFit+\varepsilon}\right) \tag{5}$$

其中，$k\left(k\neq i\right)$ 是一个[1，N]之间的随机正整数。a_1 和 a_2 是两个[0,2]的常量，$pFit_i$ 表示第 i 只鸟的最佳适应度值，$sumFit$ 表示整个种群的最佳适应度值之和。ε 是为了避免零因子错误而由计算机产生的最小常量。$mean_j$ 表示整个种群平均位置的第 j 个元素。A_1 描述的是一只鸟向鸟群中心移动时由环境引发的间接作用，A_2 描述的则是由某个具体冲突引发的直接作用。

基于规则 d)e)，鸟类会因逃避捕食等原因变动飞行中的位置，每当其飞行到一个新的位置，将在生产者和乞讨者之间作出选择，自行觅食或者跟随生产者获取食物，飞行行为中生产者和乞讨者的位置更新公式分别为

$$x_{i,j}^{t+1}=x_{i,j}^t+x_{i,j}^t\times\mathrm{rand}\,n\left(0,1\right) \tag{6}$$

$$x_{i,j}^{t+1}=x_{i,j}^t+\left(x_{k,j}^t-x_{i,j}^t\right)\times FL\times\mathrm{rand}\left(0,1\right) \tag{7}$$

其中：$\mathrm{rand}\,n\left(0,1\right)$ 表示均值为 0、标准差为 1 的高斯随机分布，$FL\in\left[0,2\right]$ 代表乞讨者跟随生产者寻找食物。FQ 表示鸟群的飞行间隔，其值是一个正整数。

2　基于动态惯性权重的鸟群优化算法

2.1　动态惯性权重

动态惯性权重是一种控制种群搜索能力和开发能力的机制，能够有效平衡局部搜索与全局搜索之间的关系，提高算法搜索速度与精度避免陷入局部最优[9]。Shi 和 Eberhart 将惯性权重引入例子群算法，有效提高了算法的局部和全局寻优能力[10]。

在 BSA 算法中飞行移动间隔 FQ 直接决定算法能否跳出局部最优点寻找到全局最优，FQ 越大，全搜索能力越强，越易跳出入局部最优；FQ 越小，局部搜索能力越强，越易陷入局部最优。在迭代前期，算法需要较强的全局搜索能力，FQ 较大为宜；随着迭代次数的增加，算法需要更强的局部搜索能力，FQ 较小为宜。

在 BSA 算法中 FQ 设置为常数不能较好的平衡两者之间的关系，为更好地提高算法的搜索性能，本文引入动态惯性权重来设定 FQ 的数值。经测试发现使用非线性惯性权重效果最好、效率最高，满足 BSA 算法前期需要较强的全局搜索能力，后期需要较强的局部搜索能力的要求，较好地平衡了搜索速度和精度。引入非线性惯性权重的 FQ 更新模式如式（8）所示。

$$FQ(i)=\mathrm{floor}\left(\left(FQ_{\max}-FQ_{\min}\right)\times\exp^3\left(-2\times\left(\frac{i}{M}\right)\right)+FQ_{\min}\right),\quad FQ_{\max}>FQ_{\min} \tag{8}$$

其中：$FQ(t)$ 代表当前代的 FQ 值，$FQ_{\max}$ 为最大惯性权重，本文取值为 15，$FQ_{\min}$ 为最小惯性权重，本文取值为 4，M 为最大迭代次数，$floor()$ 为取整函数，保证所得 FQ 为整数。

2.2 莱维飞行

莱维分布是法国数学家 Paul Levy 提出，Benoist-Madelbrot 进行描述的一种稳定分布[11]，这是一种短距离游走与小概率长距离搜索相间的随机游走模式[12]。研究表明莱维飞行现象在自然界中十分普遍，许多生物的行为都符合莱维飞行轨迹。莱维飞行在搜索过程中大部分时间会在某一范围内活动，小部分时间会进行大范围的跳跃，并且运动方向随机化[13]，这样就能保证算法跳出局部最优点，避免早熟。

本文将莱维飞行引入鸟群算法，对鸟群生产者觅食行为进行改进，提高算法活力及有效性，引入莱维飞行对生产者觅食更新模式如式（9）所示。

$$x_{i,j}^{t+1} = x_{i,j}^{t} + x_{i,j}^{t} \times Levy(\lambda) \tag{9}$$

其中：$x_{i,j}^{t+1}$ 表示第 $t+1$ 代鸟群第 i 只鸟第 j 维的值，$x_{i,j}^{t}$ 表示 t 代鸟群第 i 只鸟第 j 维的值，$Levy(\lambda)$ 是步长服从莱维分布的随机搜索向量，λ 为莱维飞行步长，取值范围为[1，3]，一般取值为 1.5.

2.3 改进的 BSA 算法流程图

改进的鸟群算法流程如图 2 所示。

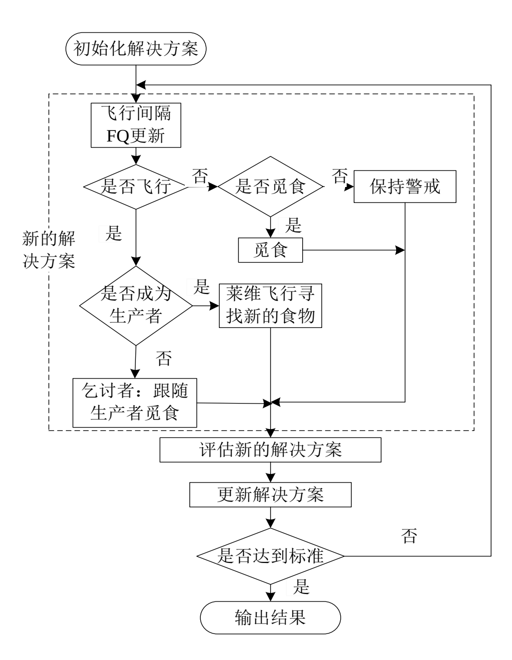


图 2　DBSA 算法流程图

3　实验结果与分析

为验证改进后的 DBSA 算法性能，本文特选取下列七个标准测试函数对 BSA、DBSA 和 PSO 算法进行对比实验。BSA、DBSA 和 PSO 算法相关参数设置如表 1 所示，标准测试函数如表 2 所示，实验对每种函数独立运行 30 次，记录实验结果如表 3 所示。

本文所选的测试函数中，除 f1、f2 函数是单峰函数外，其余 f3~f8 都是非线性的多峰函数。函数 Sphere 是一个碗状的单极值函数；函数 Trid 的搜索空间、最优解以及极值点都与搜索维度有关；Ackley 是指数函数叠加放大而得到的一个连续性实验函数，四周呈现起伏不平，中间急剧下降的状态，极为考验算法的搜索能力；函数 Dixon-Price 最优取值点与其搜索维度有关；Hartman 3-D 和 Hartman 6-D 是具有复杂系数且有多个局部最小值的函数；函数 Shekel 具有 m 个局部最小值且函数系数复杂，导致算法搜索十困难。

表 1　实验参数设置

算法	参数取值
BSA	M=100，pop=50,FQ=10 c1=c2=1.5,a1=a2=1
DBSA	M=100，pop=50,FQ=[15,4] c1=c2=1.5,a1=a2=1
PSO	ω =0.78,h1=h2=1.5

3.1 寻优结果比较

表 3 给出了 BSA、DBSA 和 PSO 算法的寻优结果，结果表明对于单峰函数 f1 而言，BSA 算法的寻优精度远高于 PSO 算法， DBSA 算法在 BSA 算法基础上进一步改善了解的质量，搜索结果更为准确；对于 f2 函数无论是 d=6 还是 d=10，BSA 和 PSO 算法都无法有效找到最优点，而 DBSA 算法则很好地获得了函数的全局最优解；对于多峰函数 f3 而言，BSA 算法寻优精度远高于 PSO 算法，DBSA 算法与 BSA 算法相比寻优精度提高约 10 个数量级；对于 f4 函数 BSA 算法寻优精度与 PSO 算法相当，DBSA 算法与 BSA 算法相比寻优精度提高约 20 个数量级；f5f6 函数是系数为矩阵形式的测试函数，对于这类函数， BSA 算法寻优误差最大，PSO 算法次之，DBSA 算法寻优精度最高；f7 函数不仅系数复杂而且最优解与 m 有关，BSA 算法在 m 不同的三种情况下都不能找到最优解；PSO 的寻优结果略优于 BSA 算法；DBSA 算法在三种情况下都能较好的找到最优解，寻优结果较 BSA 算法有显著提高。根据表 3 的统计结果显示，DBSA 算法在这 7 个函数上的搜索能力显著优于 BSA 算法和 PSO 算法。

3.2 收敛性能比较

图 3~12 给出了 BSA、DBSA 和 PSO 三种算法函数适应度随迭代次数的变化曲线，从图 3 可以看出，对于函数 f1 DSA 算法和 BSA 算法都能有较好的收敛趋势，但 DBSA 算法较 BSA 算法收敛速度更快，PSO 算法收敛最慢；从图 4~7 可以看出，PSO 和 BSA 算法对于 f2~f4 函数均不能有良好的收敛态势，BSA 算法的收敛状态优于 PSO 算法，改进后的 DBSA 算法无论在收敛速度和收敛稳定性较 BSA 算法都有显著提高且显著优于 PSO 算法；从图 8~12 可以看出 BSA 对于函数 f5~f7 均没有较好的收敛态势，不仅前期收敛速度满而且后期不能保持稳定，图 10 虽能保持稳定但不能找到全局最优解，收敛状态显著劣于 PSO 和 DBSA 算法，相较与 PSO 算法，DBSA 显然有更好的收敛速度和稳定状态。综上所述，本文所提的 DBSA 算法收敛稳定性及收敛速度都显著优于 BSA 算法和 PSO 算法。

表 2　标准测试函数

函数名	函数表达式	搜索空间	维度	理论最优解
Sphere	$f_1(x)=\sum_{i=1}^{d}x_i^2$	[-5.12,5.12]	d=20	$f\left(X^*\right)=0$
Trid	$f_2\left(x\right)=\sum_{i=1}^{d}\left(x_i-1\right)^2-\sum_{i=2}^{d}x_ix_{i-1}$	$[-d^2,d^2]$	d=6	$f\left(X^*\right)=-50$
			d=10	$f\left(X^*\right)=-210$
Ackley	$f_3\left(x\right)=20+e-\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{d}\cos\left(2\pi x_i\right)\right)$	[-32.768,32.768]	d=20	$f\left(X^*\right)=0$
Dixon-Price	$f_4\left(x\right)=\left(x_1-1\right)^2+\sum_{i=2}^{d}i\left(2x_i^2-x_{i-1}\right)^2$	[-10,10]	d=20	$f\left(X^*\right)=0$
Hartman 3-D	$f_5\left(x\right)=-\sum_{i=1}^{4}\alpha_i\exp\left(-\sum_{j=1}^{3}A_{ij}\left(x_j-P_{ij}\right)^2\right)$ $\alpha=\left(1.0,1.2,3.0,3.2\right)^T$ $A=\begin{pmatrix}3.0 & 10 & 30\\0.1 & 10 & 35\\3.0 & 10 & 30\\0.1 & 10 & 35\end{pmatrix}$ $P=10^{-4}\begin{pmatrix}3689 & 1170 & 2673\\4699 & 4387 & 7470\\1091 & 8732 & 5547\\381 & 5743 & 8828\end{pmatrix}$	(0,1)	d=3	$f\left(X^*\right)=-3.86278$
Hartman 6-D	$f_6\left(x\right)=-\sum_{i=1}^{4}\alpha_i\exp\left(-\sum_{j=1}^{6}A_{ij}\left(x_j-P_{ij}\right)^2\right)$ $\alpha=\left(1.0,\ 1.2,\ 3.0,\ 3.2\right)^T$ $A=\begin{pmatrix}10 & 3 & 17 & 3.5 & 17 & 8\\0.05 & 10 & 17 & 0.1 & 8 & 14\\3 & 3.5 & 1.7 & 10 & 17 & 8\\17 & 8 & 0.05 & 10 & 0.1 & 14\end{pmatrix}$ $P=10^{-4}\begin{pmatrix}1312 & 1696 & 5569 & 124 & 8283 & 5886\\2329 & 4135 & 8307 & 3736 & 1004 & 9991\\2348 & 1451 & 3522 & 2883 & 3047 & 6650\\4047 & 8828 & 8732 & 5743 & 1091 & 381\end{pmatrix}$	(0,1)	d=6	$f\left(X^*\right)=-3.32237$
Shekel	$f_7\left(x\right)=-\sum_{i=1}^{m}\left(\sum_{j=1}^{4}\left(x_j-C_{ji}\right)^2+\beta_i\right)^{-1}$ $\beta=\frac{1}{10}\left(1,\ 2,\ 2,\ 4,4,\ 6,\ 3,\ 7,\ 5,\ 5\right)^T$ $C=\begin{pmatrix}4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0\\4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6\\4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0\\4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6\end{pmatrix}$	[0,10]	d=4	m=5　$f\left(X^*\right)=-10.1532$
				m=7　$f\left(X^*\right)=-10.4029$
				m=10　$f\left(X^*\right)=-10.5364$

3.3　时间性比较

表 4 给出了 BSA、DBSA 和 PSO 算法迭代 100 次的运行时间对比，每种算法迭代 100 代，可以看出 DBSA 算法和 BSA 算法在运行时间上并无较大差别，较 PSO 少用时约 0.1s，这是因为 DBSA 算法虽因引入莱维飞行，计算复杂度大幅增加，但由于本文同时引入了动态惯性权重的飞行间隔更新策略，有效减少了莱维飞行次数，所以 DBSA 算法在运行时间上较 BSA 算法的增加并不明显，这说明在大幅提高 BSA 算法活力和准确性，显著提高算法收敛速度的前提下，这种改进策略是可取的。

4　结束语

针对鸟群算法在求解最优化函数中存在易陷入局部最优、收敛速度慢、求解精度低等问题，本文提出一种基于动态惯性权重的鸟群优化算法，通过引入非线性动态权重更新飞行间隔，以此分配算法局部搜索与全局搜索资源，平衡算法搜索精度与速度；引入莱维飞行来对生产者觅食策略进行修正，提高算法活力。通过对七个标准测试函数实验表明，与传统 BSA 算法相比，DBSA 算法可以有效提高搜索精度，，增强收敛速度，与 BSA

算法相比，本文所提算法在求解多维复杂函数最优化的问题上是具有竞争力的。

表 3　BSA 与 LWBSA 算法的比较结果

函数	算法		寻优结果		
			最佳寻优值	平均寻优值	最差寻优值
$f_1(x)$		BSA	5.2034e-137	4.3752e-136	1.1528e-134
		DBSA	0	1.2931e-143	2.3754e-142
		PSO	0.0032	0.0584	0.2486
$f_2(x)$	d=6	BSA	-48.3533	-30.4774	-2.4395
		DBSA	-50	-50	-50
		PSO	-47.6951	-20.1855	20.3340
	d=10	BSA	-107.6532	-83.5703	6.6080
		DBSA	-210	-209.9992	-209.9946
		PSO	178	571.2	1214
$f_3(x)$		BSA	8.8816e-16	4.7533e-4	0.2746
		DBSA	1.5099e-25	6.5027e-19	2.8031e-17
		PSO	0.0082	1.9633	3.1267
$f_4(x)$		BSA	0.0169e-2	0.0079	0.0732
		DBSA	1.06e-27	9.9781e-24	9.9435e-20
		PSO	9.2372e-2	6.6929e-1	6.6667e-1
$f_5(x)$		BSA	-3.85145	-2.97580	-2.76660
		DBSA	-3.86278	-3.86278	-3.86278
		PSO	-3.86278	-3.86275	-3.86217
$f_6(x)$		BSA	-3.06774	-2.50341	-1.05732
		DBSA	-3.32237	-3.32237	-3.32235
		PSO	-3.32237	-3.23892	-3.20316
$f_7(x)$	m=5	BSA	-9.7781	-5.4377	-2.6304
		DBSA	-10.1532	-10.1532	-10.1532
		PSO	-10.1532	-6.6767	-3.2868
	m=7	BSA	-10.3100	-5.2786	-2.7345
		DBSA	-10.4029	-10.4029	-10.4029
		PSO	-10.4029	-7.5916	-2.7519
	m=10	BSA	-9.8062	-4.4226	-2.4261
		DBSA	-10.5364	-10.5364	-10.5364
		PSO	-10.5364	-5.9940	-2.4217

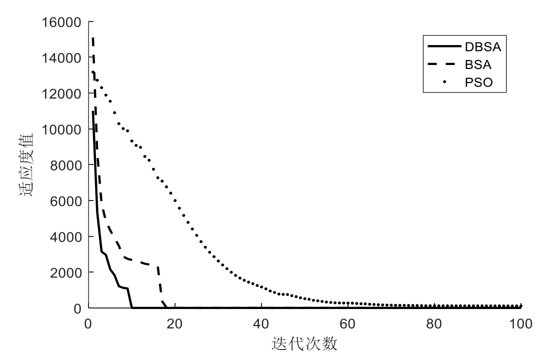


图 3　算法收敛对比图（f1）

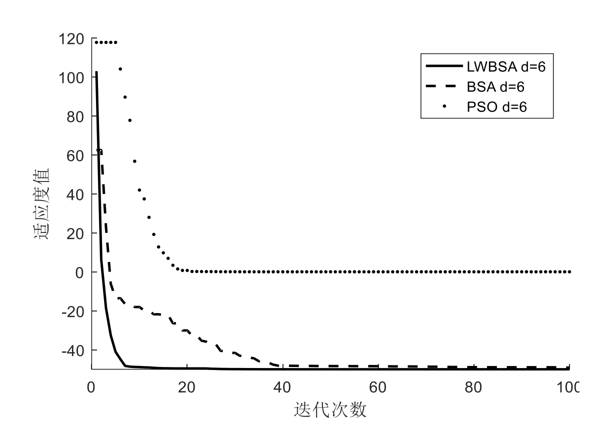


图 4　算法收敛对比图（f2 d=6）

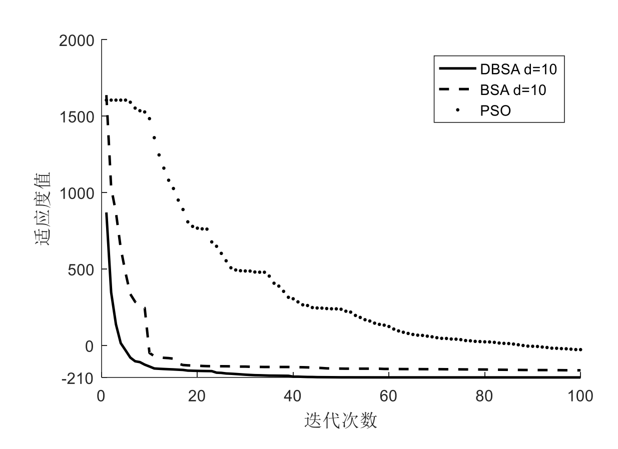


图 5　算法收敛对比图（f2 d=10）

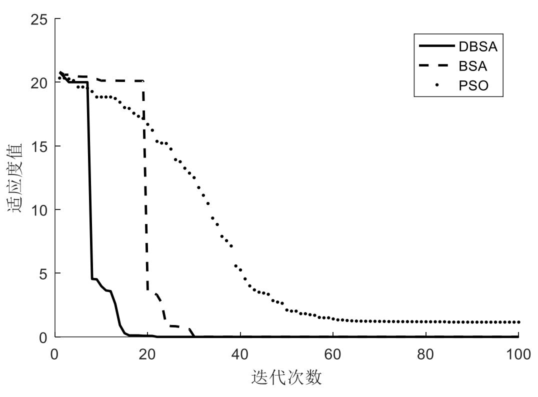


图 6　算法收敛对比图（f3）

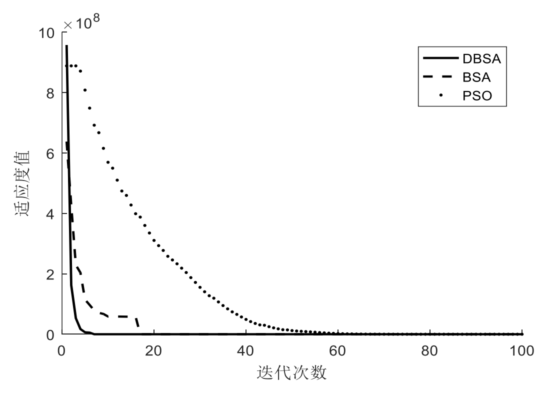


图 7　算法收敛对比图（f4）

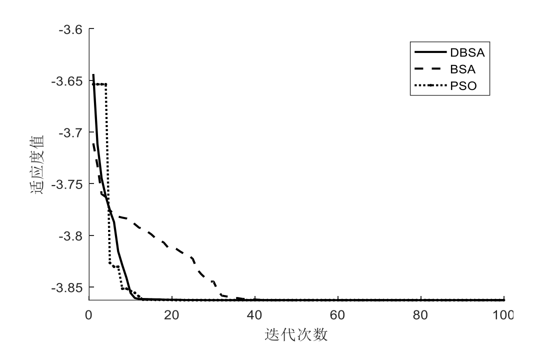


图 8 算法收敛对比图（f5）

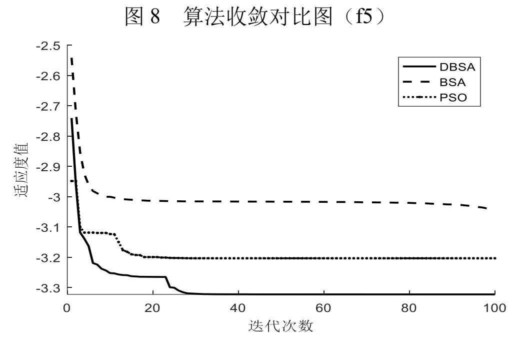


图 9 算法收敛对比图（f6）

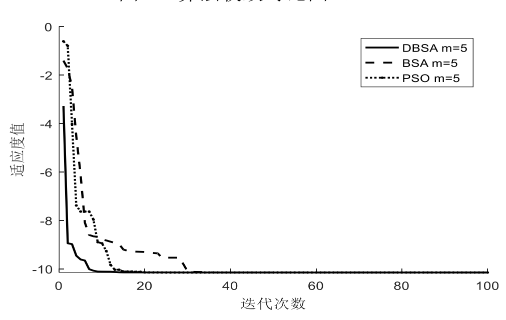


图 10 算法收敛对比图（f7 m=5）

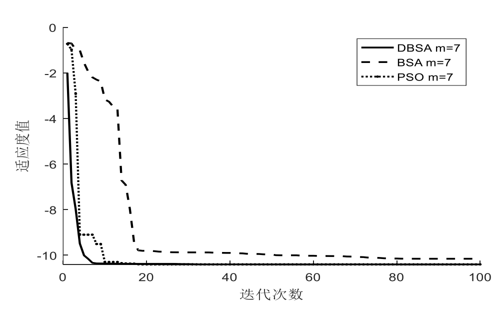


图 11 算法收敛对比图（f7 m=7）

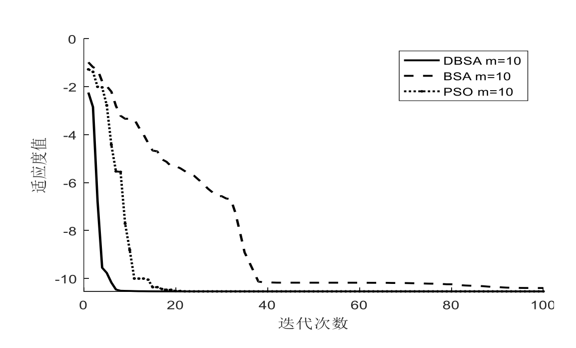


图 12 算法收敛对比图（f7 m=10）

表 4 BSA、DBSA 与 PSO 时间对比

算法	函数									
	f_1	f_2		f_3	f_4	f_5	f_6	f_7		
		d=6	d=10					m=5	m=7	m=10
BSA	0.447s	0.449s	0.496s	0.463s	0.366s	0.506s	0.493s	0.527s	0.544s	0.573s
DBSA	0.439s	0.512s	0.494s	0.569s	0.317s	0.572s	0.474s	0.458s	0.500s	0.600s
PSO	0.573s	0.603s	0.605s	0.636s	0.603s	0.719s	0.751s	0.707s	0.693s	0.696s

参考文献：

[1] Meng X B, Gao X Z, Lu L, *et al.* A new bio-inspired optimisation algorithm: bird swarm algorithm [J]. Journal of Experimental & Theoretical Artificial Intelligence, 2015: 1-15.

[2] 屈迟文，傅彦铭，罗明山，等. 求解柔性作业车间调度问题的鸟群算法 [J/OL]. 计算机工程与应用，(2017-08-24). http://kns. cnki. net/kcms/detail/11. 2127. TP. 20170824. 0913. 010. html.

[3] 曾嶒，彭春华，王奎，等. 基于鸟群算法的微电网多目标运行优化 [J]. 电力系统保护与控制, 2016, 44 (13): 117-122.

[4] 肖海军，卢常景，何凡. 基于鸟群算法的 SVM 参数选择 [J]. 中南民族大学学报：自然科学版, 2017, 36 (3): 90-94.

[5] Parashar M, Rajput S, Dubey H M, *et al.* Optimization of benchmark functions using a nature inspired bird swarm algorithm [C]// Proc of International Conference on Computational Intelligence & Communication Technology. 2017.

[6] Cui D, Jin B, Bureau W W. Improved bird swarm algorithm and its application to reservoir optimal operation [J]. Journal of China Three Gorges University, 2016.

[7] Cui D, Jin B. Application of the bird swarm algorithm-projection pursuit regression model to prediction of multivariate annual runoff [J]. Pearl River, 2016.

[8] Kn L, Reddy D B R, Kalavathi D M S. Snow finch bird swarm optimization algorithm for solving reactive power problem [J]. International Journal of Mathematical Engineering & Management Sciences, 2016.

[9] 周欢，李煜. 具有动态惯性权重的布谷鸟搜索算法 [J]. 智能系统学报, 2015, 10 (4): 645-651.

[10] Shi Y, Eberhart R. Modified particle swarm optimizer [C]// Proc of IEEE International Conference on Evolutionary Computation Proceedings. 1997: 69-73.

[11] 刘晓龙，宁芊，赵成萍，等. 基于莱维飞行的鸟群优化算法 [J]. 计算机测量与控制, 2016, 24 (12): 194-197.

[12] 王庆喜，郭晓波. 基于莱维飞行的粒子群优化算法 [J]. 计算机应用研究, 2016, 33 (9): 2588-2591.

[13] 牛海帆，宋卫平，宁爱平. 莱维飞行与粒子群的混合搜索算法 [J]. 太原科技大学学报, 2016, 37 (1): 6-11.